



Tutorial for CH4

汇编语言程序设计

第4章 习题

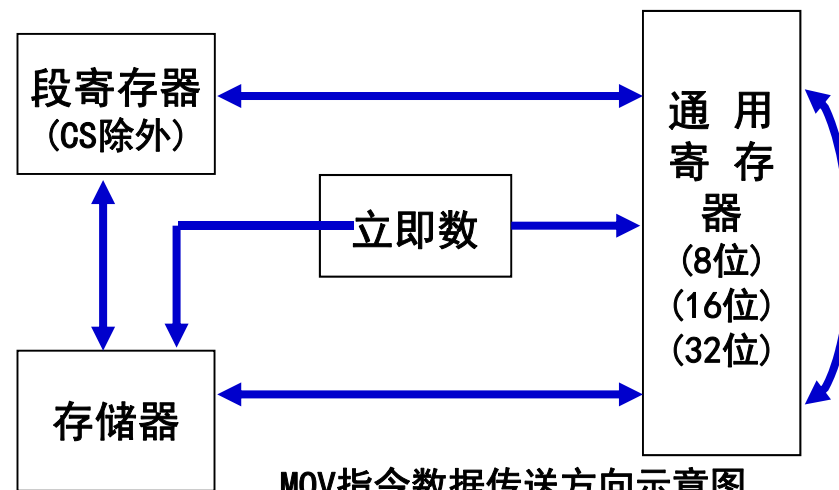
4.1 指出下列指令的错误:

- (1) **MOV** AH, BX
- (2) **MOV** [BX], [SI]
- (3) **MOV** AX, [SI][DI]
- (4) **MOV** MYDAT[BX][SI], ES:AX
- (5) **MOV BYTE PTR** [BX], 1000
- (6) **MOV** BX, **OFFSET** MYDAT[SI]
- (7) **MOV** CS, AX
- (8) **MOV** ECX, AX

相关知识 - MOV 指令 (P58)

传送指令: MOV DST, SRC

执行操作: (DST) ← (SRC)



注意:

- 立即数不能作为目标操作数
- 立即数不能直接送段寄存器 × MOV DS, 2000H
- DST不能是CS, 因为随意修改CS会引起不可预料的结果
- DST、SRC不同时为段寄存器 × MOV DS, ES
- 两个存储单元之间不能直接传送

相关知识 - 相关伪操作

➤ PTR属性操作符

- type **PTR** variable \pm constant expression (P148)
- 指定操作数的类型属性，优于隐含的属性
- 类型 type 可以是 BYTE, WORD, DWORD, FWORD, QWORD, TBYTE

➤ OFFSET (P157)

- OFFSET variable 或 label
- 汇编程序将回送变量或标号的偏移地址值

相关知识 - 8086寻址方式

与数据有关的寻址方式：以 **MOV** 指令为例

- | | |
|------------|-----------------------------------|
| • 立即寻址 | MOV AX, 3069H |
| • 寄存器寻址 | MOV AL, BH |
| • 直接寻址 | MOV AX, [2000H] |
| • 寄存器间接寻址 | MOV AX, [BX] |
| • 寄存器相对寻址 | MOV AX, COUNT [SI] |
| • 基址变址寻址 | MOV AX, [BP] [DI] |
| • 相对基址变址寻址 | MOV AX, MASK [BX] [SI] |

存储器寻址

4.1 指出下列指令的错误:

- (1) **MOV** AH, BX
- (2) **MOV** [BX], [SI]
- (3) **MOV** AX, [SI][DI]
- (4) **MOV** MYDAT[BX][SI], ES:AX
- (5) **MOV BYTE PTR** [BX], 1000
- (6) **MOV** BX, **OFFSET** MYDAT[SI]
- (7) **MOV** CS, AX
- (8) **MOV** ECX, AX

4.1 答案:

- (1) **MOV** AH, BX ; 寄存器类型不匹配
- (2) **MOV** [BX], [SI] ; 不能都是存储器操作数
- (3) **MOV** AX, [SI][DI] ; [SI]和[DI]不能一起使用
- (4) **MOV** MYDAT[BX][SI], ES:AX ; AX不能使用段超越
- (5) **MOV BYTE PTR** [BX], 1000 ; 操作数类型不匹配
- (6) **MOV** BX, **OFFSET** MYDAT[SI]; 汇编时**OFFSET**将回送变量或标号的偏移地址值
- (7) **MOV** CS, AX ; CS不能用作目的寄存器
- (8) **MOV** ECX, AX ; 操作数类型不匹配

4.2 下面哪些指令是非法的？(假设OP1, OP2是已经用DB定义的变量)

- (1) `CMP 15, BX`
- (2) `CMP OP1, 25`
- (3) `CMP OP1, OP2 ;`
- (4) `CMP AX, OP1`

相关知识： P69，算术指令使用规定。

4.2 答案

(1) **CMP** 15, BX

立即数不能作为目的操作数

(2) **CMP** OP1, 25

(3) **CMP** OP1, OP2 ;

不能都是存储器操作数

(4) **CMP** AX, OP1

类型不匹配，应为: **CMP** AX, WORD PTR OP1

4.3 假设下列指令中的所有标识符均为类型属性为字的变量，请指出下列哪些指令是非法的？它们的错误是什么？

- (1) **MOV** BP, AL
- (2) **MOV** WORD_OP1[BX+4*3][DI], SP
- (3) **MOV** WORD_OP1, WORD_OP2
- (4) **MOV** AX, WORD_OP1[DX]
- (5) **MOV** SAVE_WORD, DS
- (6) **MOV** SP, SS:DATA_WORD[BX][SI]
- (7) **MOV** [BX][SI], 2
- (8) **MOV** AX, WORD_OP1+WORD_OP2
- (9) **MOV** AX, WORD_OP1-WORD_OP2+100
- (10) **MOV** WORD_OP1, WORD_OP1-WORD_OP2

测试

4.3 答案

- (1) `MOV BP, AL` ; 寄存器类型不匹配
- (2) `MOV WORD_OP1[BX+4*3][DI], SP`
- (3) `MOV WORD_OP1, WORD_OP2` ; 不能都是存储器操作数
- (4) `MOV AX, WORD_OP1[DX]` ; `DX`不能用于存储器寻址
- (5) `MOV SAVE_WORD, DS`
- (6) `MOV SP, SS:DATA_WORD[BX][SI]`
- (7) `MOV [BX][SI], 2` ; `[BX][SI]`未指出数据类型
- (8) `MOV AX, WORD_OP1+WORD_OP2` ; 两个变量的地址不能相加
- (9) `MOV AX, WORD_OP1-WORD_OP2+100`
- (10) `MOV WORD_OP1, WORD_OP1-WORD_OP2`

课堂练习：判断下列哪些指令是错误的

- (1) **MOV** CS, DS
- (2) **MOV** AH, SI
- (3) **MOV** AH, BYTE PTR [BX][BP]
- (4) **MOV** AX, 1000
- (5) **MOV** AL, BYTE PTR [SI][DI]
- (6) **MOV** AL, WORD PTR [BX][SI]
- (7) **SHR** AX, CX
- (8) **ADD** BL, WORD PTR [SI]
- (9) **MUL** AL, BL
- (10) **DIV** BH

4.4 假设VAR1和VAR2为字变量，LAB为标号，试指出下列指令的错误之处：

(1) **ADD** VAR1, VAR2

(2) **SUB** AL, VAR1

(3) **JMP** LAB[SI]

(4) **JNZ** VAR1

(5) **JMP** NEAR LAB

4.4 假设VAR1和VAR2为字变量，LAB为标号，试指出下列指令的错误之处：

- (1) **ADD** VAR1, VAR2 ; 不能都是存储器操作数
- (2) **SUB** AL, VAR1 ; 数据类型不匹配
- (3) **JMP** LAB[SI] ; LAB是标号而不是变量名，后面不能加[SI]
- (4) **JNZ** VAR1 ; VAR1是变量而不是标号
- (5) **JMP** NEAR LAB ; 应使用NEAR PTR

测试

4.5 画图说明下列语句所分配的存储空间及初始化的数据值

➤ (略)

➤ 12, -12 补码问题

4.7 请设置一个数据段DATASG，其中定义以下字符变量或数据变量

- (1) FLD1B为字符串变量： 'personal computer';
- (2) FLD2B为十进制数字字节变量： 32;
- (3) FLD3B为十六进制数字字节变量： 20;
- (4) FLD4B为二进制数字字节变量： 01011001;
- (5) FLD5B为数字的ASCII字符字节变量： 32654;
- (6) FLD6B为10个零的字节变量;
- (7) FLD7B为零件名(ASCII码)及其数量(十进制数)的表格:

PART1 20

PART2 50

PART3 14

- (8) FLD1W为十六进制数字变量： FFF0;
- (9) FLD2W为二进制数的字变量： 01011001;
- (10) FLD3W为(7)零件表的地址变量;
- (11) FLD4W为包括5个十进制数的字变量： 5, 6, 7, 8, 9;
- (12) FLD5W为5个零的字变量;
- (13) FLD6W为本段中字数据变量和字节数据变量之间的地址差。

DATASG **SEGMENT**

FLD1B DB 'personal computer'

FLD2B DB 32

FLD3B DB 20H

FLD4B DB 01011001B

FLD5B DB '32654'

FLD6B DB 10 DUP (0)

FLD7B DB 'PART1', 20

DB 'PART2', 50

DB 'PART3', 14

FLD1W DW 0FFF0H

FLD2W DW 01011001B

FLD3W DW FLD7B

FLD4W DW 5, 6, 7, 8, 9

FLD5W DW 5 DUP (0)

FLD6W DW FLD1W - FLD1B

DATASG **ENDS**

(测试)

4.9 有符号定义语句如下:

BUFF DB 1, 2, 3, '123'

EBUFF DB 0

L EQU EBUFF - BUFF

问L的值是多少?

(EQU 伪操作 P149)

4.10 假设程序中的数据定义如下：

LNAME DB 30 DUP (?)

ADDRESS DB 30 DUP (?)

CITY DB 15 DUP (?)

CODE_LIST DB 1, 7, 8, 3, 2

- (1) 用一条MOV指令将LNAME的偏移地址放入AX。
- (2) 用一条指令将CODE_LIST的头两个字节的内容放入SI。
- (3) 用一条伪操作使CODE_LENGTH的值等于CODE_LIST域的实际长度。

4.10 答案

(1) **MOV** AX, **OFFSET** LNAME

(2) **MOV** SI, **WORD PTR** CODE_LIST

(3) CODE_LENGTH **EQU** \$ - CODE_LIST

（此语句必须放在CODE_LIST语句之后）

4.12 给出等值语句如下，下列表达式的值是多少？

等值语句：

ALPHA EQU 100

BETA EQU 25

GAMMA EQU 2

表达式：

(1) ALPHA * 100 + BETA ; =2729H

(2) ALPHA MOD GAMMA + BETA ; =19H

(3) (ALPHA + 2) * BETA - 2 ; =9F4H

(4) (BETA / 3) MOD 5 ; =3H

(5) (ALPHA + 3) * (BETA MOD GAMMA) ; =67H

(6) ALPHA GE GAMMA ; =0FFFFH

(7) BETA AND 7 ; =01H

(8) GAMMA OR 3 ; =03H

相关知识 – 表达式（p154）

- 操作数项可以由表达式组成；
- 表达式：常数、寄存器、标号、变量与一些操作符相组合的序列
- 在汇编期间，汇编程序按照一定优先级规则对表达式进行计算后可以得到一个数值或一个地址

(举例)

- 操作符以及优先级顺序（P154 ~ P159）

4.13 对于下面的数据定义，三条MOV指令分别汇编成什么？(可用立即数方式表示)

TABLEA DW 10 DUP (?)

TABLEB DB 10 DUP (?)

TABLEC DB '1234'

⋮

MOV AX, LENGTH TABLEA ; 汇编成MOV AX, 000AH

MOV BL, LENGTH TABLEB ; 汇编成MOV BL, 000AH

MOV CL, LENGTH TABLEC ; 汇编成MOV CL, 0001H

4.14 对于下面的数据定义，各条MOV指令单独执行后，有关寄存器的内容是什么？

FLDB DB ?

TABLEA DW 20 DUP (?)

TABLEB DB 'ABCD'

- (1) MOV AX, TYPE FLDB ; (AX)=0001H
- (2) MOV AX, TYPE TABLEA ; (AX)=0002H
- (3) MOV CX, LENGTH TABLEA ; (CX)=0014H
- (4) MOV DX, SIZE TABLEA ; (DX)=0028H
- (5) MOV CX, LENGTH TABLEB ; (CX)=0001H

4.21 试说明下述指令中哪些需要加上PTR操作符:

BVAL DB 10H, 20H

WVAL DW 1000H

(1) MOV AL, BVAL

(2) MOV DL, [BX]

(3) SUB [BX], 2 ;

需要, SUB BYTE PTR [BX], 2

(4) MOV CL, WVAL ;

需要, MOV CL, BYTE PTR WVAL

(5) ADD AL, BVAL+1

伪操作（伪指令）

- 数据定义伪指令
- 操作数运算符
 - 算术运算符
 - 逻辑运算符
 - 关系运算符
 - 数值回送运算符
 - 属性运算符
 - 移位运算符
- 段定义伪指令
- 简化的段定义

数据定义伪操作

[变量名] 定义符 操作数项表

执行的操作：为变量分配存储单元并将初值置入相应单元中。

其中：定义符可为以下几种：

- DB 定义字节。
- DW 定义字(2 个字节)。
- DD 定义双字(4 个字节)。
- DF 定义六字节。
- DQ 定义八字节。
- DT 定义十字节。

操作数项表可为以下几种：

- 常数或表达式。
- 字符串。
- ? 表达式。
- 带 DUP 表达式。

算术运算符

运 算 符	格 式	运 算
+	+ 表达式	取表达式正值
-	- 表达式	取表达式负值
+	表达式 1 + 表达式 2	求表达式 1、表达式 2 之和
-	表达式 1 - 表达式 2	求表达式 1、表达式 2 之差
*	表达式 1 * 表达式 2	求表达式 1、表达式 2 之积
/	表达式 1 / 表达式 2	求表达式 1、表达式 2 之商
MOD	表达式 1 MOD 表达式 2	取余数

(1) 用于数值表达式或地址表达式

(2) 用于地址表达式时，只有其结果有物理意义时才有效

常用的格式：地址 + 数字常量

地址 - 数字常量

MOV AL, 20 + 30

MOV BH, 100 - 2

MOV CX, 50 / 2

MOV AL, 80 MOD 15

逻辑运算符

➤ 逻辑运算符

运 算 符	格 式	运 算
NOT	NOT 表达式	按位取反
AND	表达式 1 AND 表达式 2	按位“与”运算
OR	表达式 1 OR 表达式 2	按位“或”运算
XOR	表达式 1 XOR 表达式 2	按位“异或”运算

➤ 逻辑运算符与逻辑指令的区别

	逻辑运算指令	逻辑运算符
出现在语句的位置	指令助记符	操作数字段
操作对象	可以是寄存器或存储器操作数	只能是整型常数
何时计算	程序运行时	源程序汇编时

➤ 举例

MOV AH, NOT 0F0H

MOV BL 40H OR 0B4H

MOV CH 50H AND 30H

关系运算符

运 算 符	格 式	运 算
EQ	表达式 1EQ 表达式 2	表达式 1=表达式 2 为真
NE	表达式 1NE 表达式 2	表达式 1<>表达式 2 为真
LT	表达式 1LT 表达式 2	表达式 1<表达式 2 为真
LE	表达式 1LE 表达式 2	表达式 1<=表达式 2 为真
GT	表达式 1GT 表达式 2	表达式 1>表达式 2 为真
GE	表达式 1GE 表达式 2	表达式 1>=表达式 2 为真

关系运算符的两个操作数必须都是常量或者同一段内的符号地址。

比较结果：0FFFFH (真), 0(假)

数值回送运算符

➤ 地址回送运算符

符 号	作用对象	执行的操作
SEG	变量或标号	返回其所在段的段基址值
OFFSET	变量或标号	返回其在段内的偏移量

➤ 符号特征回送运算符

符 号	作用对象	执行的操作
TYPE	变量或标号	用数字表示其类型属性
LENGTH	变量	用 DUP 重复定义符定义的变量, 返回分配的元素个数; 其他形式, 则返回 1
SIZE	变量	回送 LENGTH * TYPE 之积

➤ TYPE运算符

	类型属性	运算结果
变量	DB	1
	DW	2
	DD	4
	DQ	8
	DT	10
标号	NEAR	-1
	FAR	-2

属性运算符

(1) PTR

格式：类型 PTR 表达式

执行的操作：对存储器表达式，类型可为 BYTE、WORD、DWORD、QWORD 和 TBYTE；表达式为标号时，类型可为 NEAR 或 FAR。

(2) SHORT

格式：JMP SHORT 标号

执行的操作：转移的距离属性为短，即转移范围为 $-128 \sim +127$ 字节。

(3) THIS

格式：THIS 类型

执行的操作：为相应的变量名或标号指定属性，使其与下一条邻接语句的变量或标号的属性相同。

(4) HIGH/LOW

格式：HIGH/LOW 表达式

执行的操作：对表达式所对应的数值常量或存储单元偏移量进行字节分离，HIGH 返回高八位，LOW 返回低八位。

移位运算符 (SHL、SHR)

格式：表达式 SHL COUNT ; 左移

表达式 SHR COUNT ; 右移

执行的操作：表达式左移/右移 COUNT 次(COUNT 为一立即数)。

MOV AX, 410AH SHL 4

MOV BL, 0FAH SHR 2